CS602 Module1 Assignment

General Rules for Homework Assignments

- You are strongly encouraged to add comments throughout the program. Doing so will help your instructor to understand your programming logic and grade you more accurately.
- You must work on your assignments individually. You are not allowed to copy the answers from the others.
- Each assignment has a strict deadline. Late submissions will carry a penalty.
- When the term *lastName* is referenced in an assignment, please replace it with your last name.

Download and extract the starter template zip file, CS602_HW1_*lastName. Rename the folder with your last name.* Complete the following programs in this folder.

Part 1 – Node.js Modules (30 Points)

Complete the provided Node.js module, `zipCodeModule_v1.js`, with the following functionality. The module maintains an array of JavaScript objects, ***data***, as shown below. The _id property is the zip code of the city. The *pop* property is the population of the city. The *city* and *state* properties are self-explanatory.

```
[
  {"_id":"01001","city":"AGAWAM","pop":15338,"state":"MA"},
  {"_id":"01002","city":"CUSHMAN","pop":36963,"state":"MA"},
  {"_id":"01005","city":"BARRE","pop":4546,"state":"MA"},

…

   {"_id":"99927","city":"POINT BAKER","pop":426,"state":"AK"},
   {"_id":"99929","city":"WRANGELL","pop":2573,"state":"AK"},
   {"_id":"99950","city":"KETCHIKAN","pop":422,"state":"AK"}
]
```

The module should export the functions `lookupByZipCode(zip)`, `lookupByCityState(city,state)`, and `getPopulationByState(state)`.

- The function `lookupByZipCode` should return the JavaScript object from the data whose `_id` value matches the specified argument. If the specified `argument` is not present, `undefined` is returned.
- The function `lookupByCityState` should return a JavaScript object with all the matching data as shown in the outputs below.
- The function `getPopulationByState` returns the total population for the specified state as shown in the outputs below.

Write the code for the module using **only** JavaScript **for** loops and **if** statements and **without** using any Array methods like find, filter, map, reduce, etc.

Now, complete the application, hw1a.js, using the functionality of the above module. Write the code to do the following:

- Lookup by zip code: 02215, and print the results.
- Lookup by zip code: 99999, and print the results.
- Lookup by city, state: BOSTON, MA, and print the results.
- Lookup by city, state: BOSTON, TX, and print the results.
- Lookup by city, state: BOSTON, AK, and print the results.
- Get population by state: MA, and print the results.
- Get population by state: TX, and print the results.
- Get population by state: AA, and print the results.

The sample output of the application is shown below. You can optionally use the `colors` module for colors in the output.

```
> node hw1a.js

Lookup by zip code (02215)
{ _id: '02215', city: 'BOSTON', pop: 17769, state: 'MA' }

Lookup by zip code (99999)
undefined

Lookup by city (BOSTON, MA)
{
  city: 'BOSTON',
  state: 'MA',
  data: [
    { zip: '02108', pop: 3697 },
    { zip: '02109', pop: 3926 },
    { zip: '02110', pop: 957 },
    { zip: '02111', pop: 3759 },
    { zip: '02113', pop: 6698 },
    { zip: '02114', pop: 10246 },
    { zip: '02115', pop: 25597 },
    { zip: '02116', pop: 17459 },
    { zip: '02199', pop: 886 },
    { zip: '02210', pop: 308 },
    { zip: '02215', pop: 17769 }
  ]
}

Lookup by city (BOSTON, TX)
{ city: 'BOSTON', state: 'TX', data: [ { zip: '75570', pop: 7801 } ] }

Lookup by city (BOSTON, AK)
{ city: 'BOSTON', state: 'AK', data: [] }

Get Population by State (MA)
{ state: 'MA', pop: 6016425 }

Get Population by State (TX)
{ state: 'TX', pop: 16984601 }

Get Population by State (AA)
{ state: 'AA', pop: 0 }
```

Part 2 – Node.js Modules (40 Points)

Complete a different implementation of above zipCode module using the JavaScript Array specific functions. This module should be written in the `zipCodeModule_v2.js`. Only the following should be used for each function:

- `lookupByZipCode` – The JavaScript Array *find* method only.
- `lookupByCityState` – The JavaScript Array *filter* method followed by the JavaScript Array *map* method only.
- `getPopulationByState` – The JavaScript Array *reduce* method only along with one conditional.

Now, complete the application, hw1b.js, using the functionality of the above variation of the zipCode module. The output should be the same as in the Part 1.

```
> node hw1b.js

Lookup by zip code (02215)
{ _id: '02215', city: 'BOSTON', pop: 17769, state: 'MA' }

Lookup by zip code (99999)
undefined

Lookup by city (BOSTON, MA)
{
  city: 'BOSTON',
  state: 'MA',
  data: [
    { zip: '02108', pop: 3697 },
    { zip: '02109', pop: 3926 },
    { zip: '02110', pop: 957 },
    { zip: '02111', pop: 3759 },
    { zip: '02113', pop: 6698 },
    { zip: '02114', pop: 10246 },
    { zip: '02115', pop: 25597 },
    { zip: '02116', pop: 17459 },
    { zip: '02199', pop: 886 },
    { zip: '02210', pop: 308 },
    { zip: '02215', pop: 17769 }
  ]
}

Lookup by city (BOSTON, TX)
{ city: 'BOSTON', state: 'TX', data: [ { zip: '75570', pop: 7801 } ] }

Lookup by city (BOSTON, AK)
{ city: 'BOSTON', state: 'AK', data: [] }

Get Population by State (MA)
{ state: 'MA', pop: 6016425 }

Get Population by State (TX)
{ state: 'TX', pop: 16984601 }

Get Population by State (AA)
{ state: 'AA', pop: 0 }
```

Part 3 – Node.js Events (30 Points)

Complete the Node.js module, `zipCodeEmitter.js`, with the following functionality. The `ZipCodeEmitter` class inherits from the `EventEmitter`. Provide the member functions `lookupByZipCode`, `lookupByCityState`, and `getPopulationByState` for the `ZipCodeEmitter` class. These methods do not return any results. Instead, the last line in these methods should `emit` the respective event (same name as the function) along with the result of the function. The rest of the code in each of the functions should be the same as in Part1 (or Part2). From the module, export one property `ZipCodeEmitter` referencing the class.

Now, complete the application, hw1c.js, using the functionality of the above module. Write the code to do the following:

- Create the `ZipCodeEmitter` object using the default constructor.
- Write four event handlers for the three events that could be emitted by the three functions, one handler each for *lookupByZipCode* event and *getPopulationByState* event, and two handlers for *lookupByCityState* event. See the sample output for the behavior of these handlers.
- Now, using the `ZipCodeEmitter` object, do the following operations.
    - Lookup by zip code: 02215.
    - Lookup by city, state: BOSTON, MA.
    - Get population by state: MA.

Note that the above three function calls will return *undefined* as the result of the function call. The actual output shown below comes from the respective event handlers.

The sample output of the application is shown below. You can optionally use the `colors` module for colors in the output.

```
> node hw1c.js

Lookup by zip code (02215)

Event lookupByZipCode raised!
 { _id: '02215', city: 'BOSTON', pop: 17769, state: 'MA' }

Lookup by city (BOSTON, MA)

Event lookupByCityState raised! (Handler1)
{
  city: 'BOSTON',
  state: 'MA',
  data: [
    { zip: '02108', pop: 3697 },
    { zip: '02109', pop: 3926 },
    { zip: '02110', pop: 957 },
    { zip: '02111', pop: 3759 },
    { zip: '02113', pop: 6698 },
    { zip: '02114', pop: 10246 },
    { zip: '02115', pop: 25597 },
    { zip: '02116', pop: 17459 },
    { zip: '02199', pop: 886 },
    { zip: '02210', pop: 308 },
    { zip: '02215', pop: 17769 }
  ]
}

Event lookupByCityState raised! (Handler2)
 City: BOSTON, State: MA
   02108 has population of 3697
   02109 has population of 3926
   02110 has population of 957
   02111 has population of 3759
   02113 has population of 6698
   02114 has population of 10246
   02115 has population of 25597
   02116 has population of 17459
   02199 has population of 886
   02210 has population of 308
   02215 has population of 17769

Get Population by State (MA)
Event getPopulationByState raised!
 { state: 'MA', pop: 6016425 }
```

Submission: Export your CS602_HW1_*lastName* folder containing all the relevant files as a zip file, and upload the zip file to the Assignment section. Please make sure that lastName in the folder is indeed your Last Name.